N91-19012

1990

NASA/ASEE SUMMER FACULTY FELLOWSHIP PROGRAM

MARSHALL SPACE FLIGHT CENTER
THE UNIVERSITY OF ALABAMA

A HIGH-FIDELITY SATELLITE EPHEMERIS PROGRAM FOR EARTH SATELLITES IN ECCENTRIC ORBITS

Prepared By:	David R. Simmons
Academic Rank:	Professor
University and Department:	Louisiana College Mathematics and Computer Science
NASA/MSFC:	
Laboratory:	Systems Analysis and Integration
Division:	Mission Analysis
Branch:	Flight Mechanics
MSFC Colleague:	Dr. Larry Mullins

Contract Number:	NGT-01-002-099 The University of Alabama

Description of the Project

The Flight Mechanics Branch is currently using a program for mission planning called the Analytic Satellite Ephemeris Program (ASEP). This program, written by Jim McCarter, produces projected data for orbits that remain fairly close to the Earth; ASEP does not take into account lunar and solar perturbations. These perturbations are accounted for in another program called GRAVE, written several years ago by Roger Burrows. This project is a revision of GRAVE which incorporates more flexible means of input for initial data, provides additional kinds of output information, and makes use of structured programming techniques to make the program more understandable and reliable.

Project Goals

Dr. Mullins and I set these goals for the project:

(1) To make the program easier to use, particularly in the initialization phase.

(2) To make the program easier to understand, and hence, easier to maintain and more reliable.

(3) To include output concerning lighting conditions.

(4) To calculate Earth shadow entry and exit times.

(5) To calculate bearings from ground stations.

Personal Goals

I had set these personal goals for the Summer Faculty Fellowship Program:

(1) To learn more about celestial mechanics.

(2) To exercise my FORTRAN programming skills and learn new features of FORTRAN.

(3) To apply modern structured programming techniques to a FORTRAN program.

(4) To get experience with a non-academic work environment.

(5) To develop a high-fidelity ephemeris program in Pascal to be used on a Personal Computer.

(6) To learn more about NASA and the space program.

(7) To have a good, intellectually refreshing summer.

(8) To get to know new people.

ORBIT

The resulting program is called ORBIT. ORBIT is written in FORTRAN; the SAIL1 VAX system was used to develop the program. In keeping with structured programming concepts, the program is divided into numerous sub-programs, each with a well-defined task to perform. The text of the source code for one sub-program can usually be printed on a page or less. Most of the variable names are whole words or short phrases which clearly identify the nature of the variable and its role in the program.

ORBIT: Initialization

ORBIT is divided into three major qhases: initialization, integration, and output. During the linking process, the block data subprogram, Load Common, gives initial values to the key variables in COMMON. Later during the initialization phase, the Get Parameter subroutine uses tree-structured menus to give users an opportunity to change the starting and ending times, output defaults and state vectors. Get Parameter can change any of the three forms of state vector (cartesian, spherical-polar, and osculating orbital elements) that are used in the program; the other forms are always re-calculated to conform to the new one. Get Parameter also provides for the selection of the kind of output to be provided.

ORBIT: Integration

The Integration phase of the program calculates new values for the elapsed time and the cartesian state vector describing the motion of the satellite. This section of the program follows the GRAVE program very closely. The Encke method is used; a subroutine called COAST calculates a position along the osculating ellipse from the current position; this position is used by the subroutine DEQG for the calculation of both gravitational and atmospheric forces. DEQG is called by RKG, a general routine for solving first-order differential equations; RKG uses Fehlberg's 13 Step version of the Runge-Kutta method. RKG is used in ORBIT with no change from its previous form. COAST has a very tangled structure; I had to split it into subroutines bases on its syntax rather than on its meaning. On the other hand, DEQG has been split into sub-programs in a natural and well-structured way.

ORBIT: Output
ORBIT produces a complete set of output data before the beginning of the integration, and after the end. The uses of the program can select an option to generate time and state-vector, which is assumed to be relative to an inertial reference system. A spherical-polar state vector and a set of osculating elements are calculated for the output phase by a set of routines organized through a master routine called UpDate Common. These subroutines also adjust various lunar, solar, and time-related variables that are maintained in COMMON. A set of routines controlled by a subroutine called Report State then display those values and calculate other values which are also displayed. Another user option is to have the displayed values stored in a file.

Pre-existing Subroutines
ORBIT makes either direct or indirect use of about a score of special-purpose routines already available in the MSFC computer systems, along with modified versions of the DEQG and COAST routines from the GRAVE program. It would not have been possible to complete this project in ten weeks if all these routines had not veen available. The use of these routines should also make it easier for MSFC personnel who are already familiar with them to understand this program.

Results: Project Goals

(1) The menu-driven input section of ORBIT should make it easy to use; some improvements can still be made.

(2) The structured nature of the program should make it easier to understand; it should be possible to find the section of ORBIT which performs a particular task and either change or add to it.

(3,4,5) With formulas provided by Larry Mullins, I was able to include output concerning lighting conditions, Earth shadow entry and exit times, and bearings from ground stations.

Results: Personal Goals

(1) I worked through MSFC Course 4181, General Description of Orbits, and was able to spend some time reading in Danby's book. I learned a lot about celestial mechanics.

(2) I exercised my FORTRAN programming skills enough that in the last few weeks I only needed to consult a text about once a week. I learned to use BLOCK IF, BLOCK DO, and DO WHILE statements in FORTRAN.

Results: Personal Goals

(3) It is definitely possible to do structured programming in FORTRAN. the new parts of ORBIT are quite structured; some of the old parts still have GO TO statements and remnants of the old style of organization.

(4) The work environment at MSFC is surprisingly like that of a college, except for the lack of classes. The kind of conversations, the paperwork, and the intellectual atmosphere are similar in many ways.

(5) There was little time to work on a Pascal version of an ephemeris program to be used on a Personal Computer. I copied the FORTRAN files onto a floppy disk in MSDOS format and was able to get several Pascal procedures working, including RKG. This is something to work on during the next year.

(6) Frank Six and Mike Freeman organized an excellent series of seminars and tours to help us learn more about NASA and the space program.

(7) This summer has been relaxing, stimulating, and highly educational.

(8) Sharing an office with John Aberg has been an excellent way to get to know new people.

Additional work needed

A great deal of re-structuring can still be done on the program. There is a need for program output on the occurrence of specific orbital events. File output organized for conversion into graphic output would be very helpful.

References

Roger Burrows, MSFC: GRAVE.FOR (1985-1988)
J.M.A. Danby: Fundamentals of Celestial Mechanics, 2nd ed.
Larry Mullins, MSFC Course 4181: Gen. Description of Orbits

Acknowledgments

I am grateful for the opportunities provided by NASA, MSFC, ASEE, and the University of Alabama through the Summer Faculty Fellowship Program. Dr. Larry Mullins has been extremely helpful; he has become a good friend, as has John Aberg. Mike Freeman and Frank Six have organized the program very well. Jim Lomas set up my terminal and helped me transfer files to MSDOS; Steve Deutchendorf patiently helped me with the VAX. Thank you.